

Appendix A

```
import java.awt.*;
import java.applet.*;
import java.net.*;
import java.io.*;
import java.awt.image.*;

class TuneGenerator extends InputStream
{
    public static final int AU=0,WAV=1,AUSAMPLERATE=8012;
    private static final byte EXPONENT[] = {
        0,0,1,1,2,2,2,2,3,3,3,3,3,3,3,3,
        4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,
        5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,
        5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,
        6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
        6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
        6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
        6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
        7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
        7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
        7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
        7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
        7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
        7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
        7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
        7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7};
    private static final int BIAS = 0x84;
    private static final int CLIP = 32635;

    int count = 0;
    int tone = 4000;
    int volume = 100;
    int x=0;

    private byte toMu(int sample){
        int sign;
        if (sample < 0) {
            sample = -sample;
            sign = 0x80;
        } else {
            sign = 0;
        }
        if (sample>CLIP)
            sample=CLIP;
        sample = sample + BIAS;
        int exponent=EXPONENT[(sample>>7) & 0xff];
        int mantissa=((sample>> (exponent+3)) & 0x0f);
        int ulawbyte=((~(sign | (exponent << 4) | mantissa))&0xff);
    }
}
```

```

        if (ulawbyte==0)                ulawbyte = 2;
        return (byte)ulawbyte;
    }

    public int read(byte arr[], int offset, int len)
    {
        int ret = -1;

        try {
            ++count;

            if (count > 2)
            {
                int i, j;

                for (i=0; i<len; i++, x++)
                {
                    double val = Math.sin((double)x /
(double)AUSAMPLERATE * tone) * (double)volume * 100;
                    val += Math.sin((double)x / (double)AUSAMPLERATE *
2 * tone) * (double)volume * 100;

                    arr[i] = toMu((int)val);
                }

                ret = len;
            }
        } catch (Exception e)
        {}

        return len;
    }

    public int read()
    {
        byte arr[] = new byte[1];
        read(arr, 0, 1);
        return arr[0];
    }

    public void setTone(int t)
    {
        tone = (int)( (double)tone * .9 + .1 * (double)t );
    }

    public void setVolume(int v)

```

```

    {
        volume = v;
    }
}

```

```

public class Synth extends Applet

```

```

{
    TuneGenerator is = null;
    URLConnection uc;

    public void init()
    {
        try {

            is = new TuneGenerator();
            sun.audio.AudioPlayer.player.start(is);

        } catch (Exception e) {}
    }

    public boolean handleEvent(Event event)
    {
        is.setTone(event.x * 10);
        is.setVolume(event.y * 10);
        return super.handleEvent(event);
    }
}

```

0062T-8949260

Appendix B

```
<applet code=Synth.class name=Synth width=320 height=200>  
<param name=label value="This string was passed from the HTML host.">  
<param name=background value="008080">  
<param name=foreground value="FFFFFF">  
<applet>
```

006277" 89492260